

FIG. 1

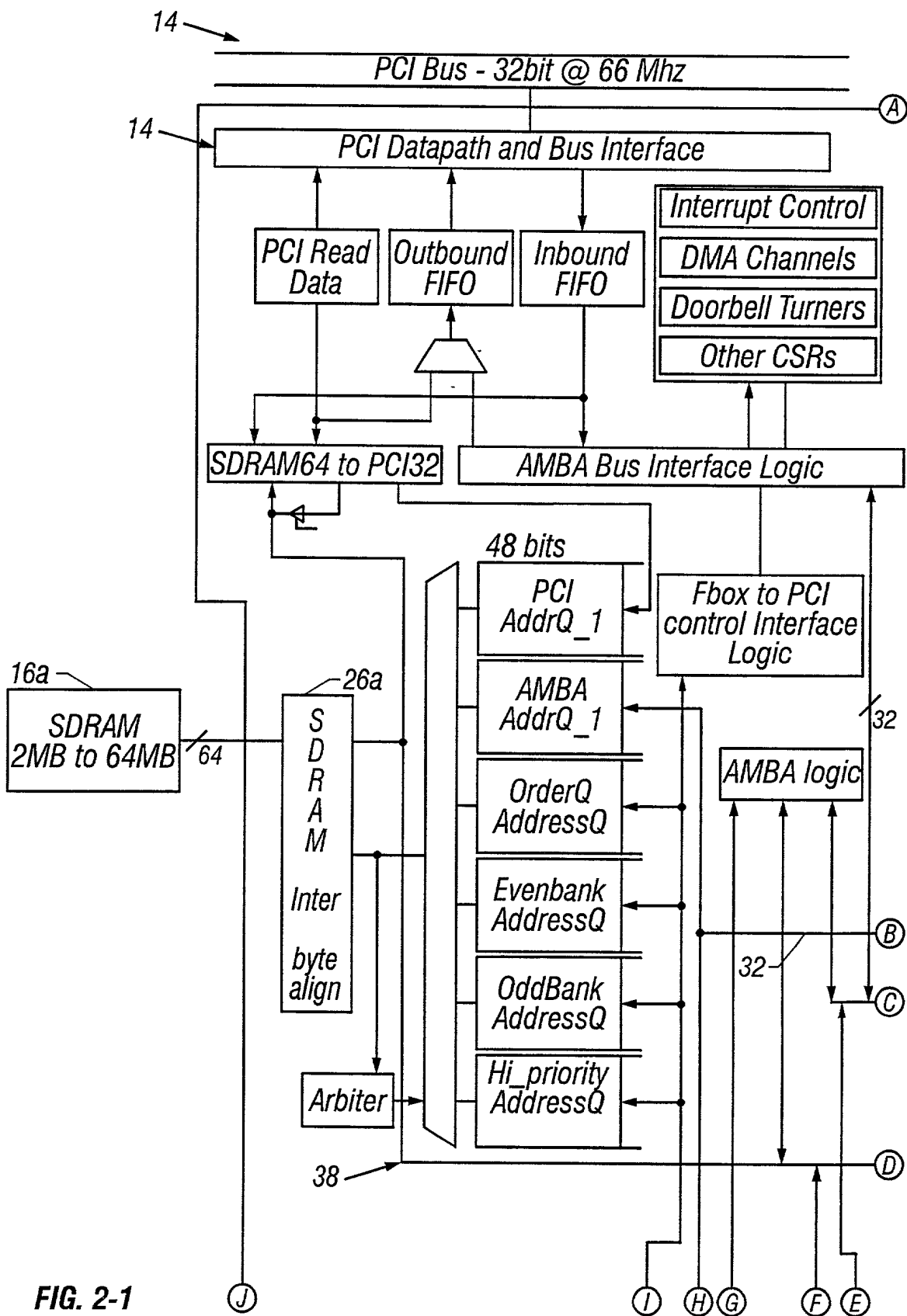


FIG. 2-1

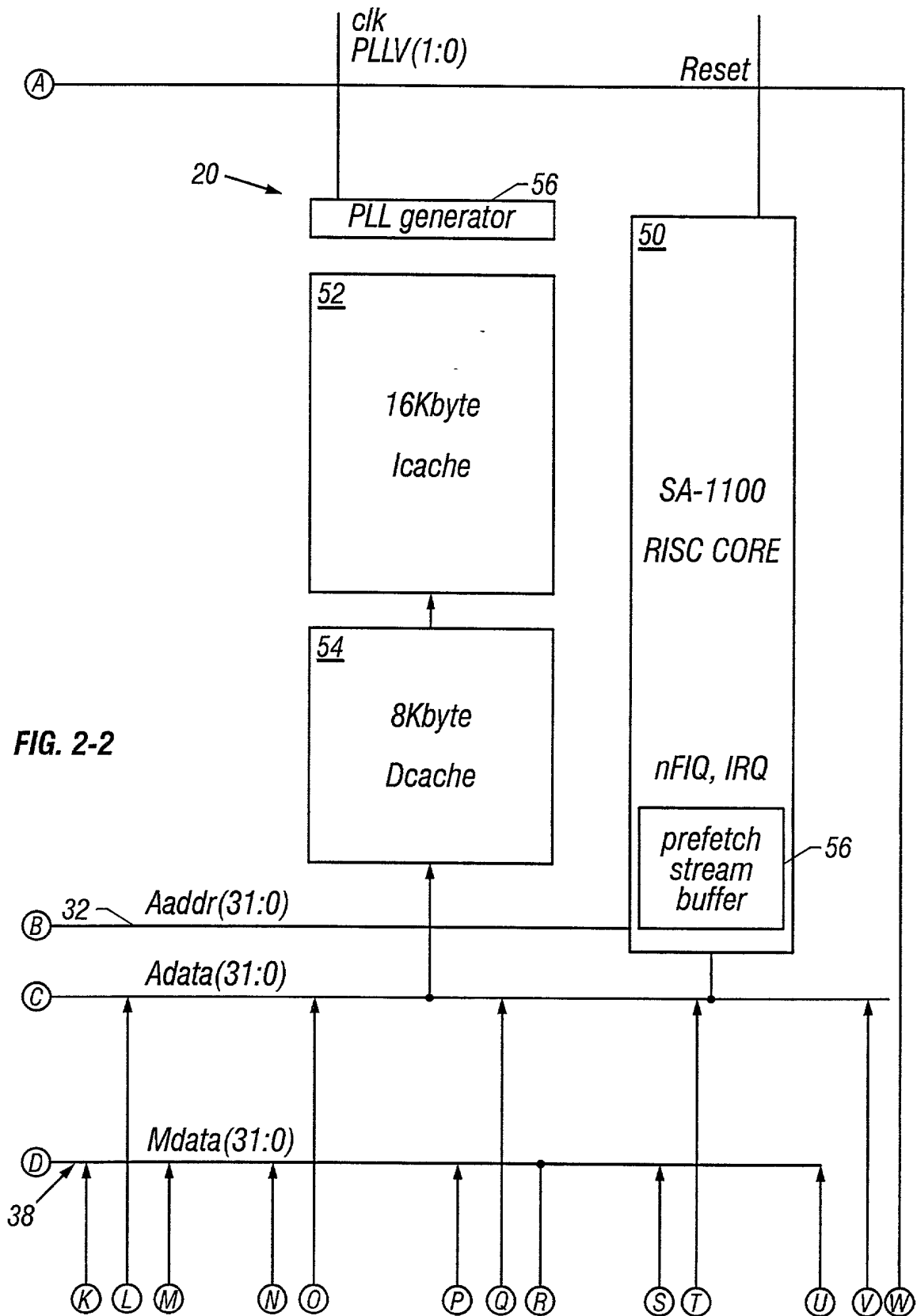


FIG. 2-2

T06250"49029860

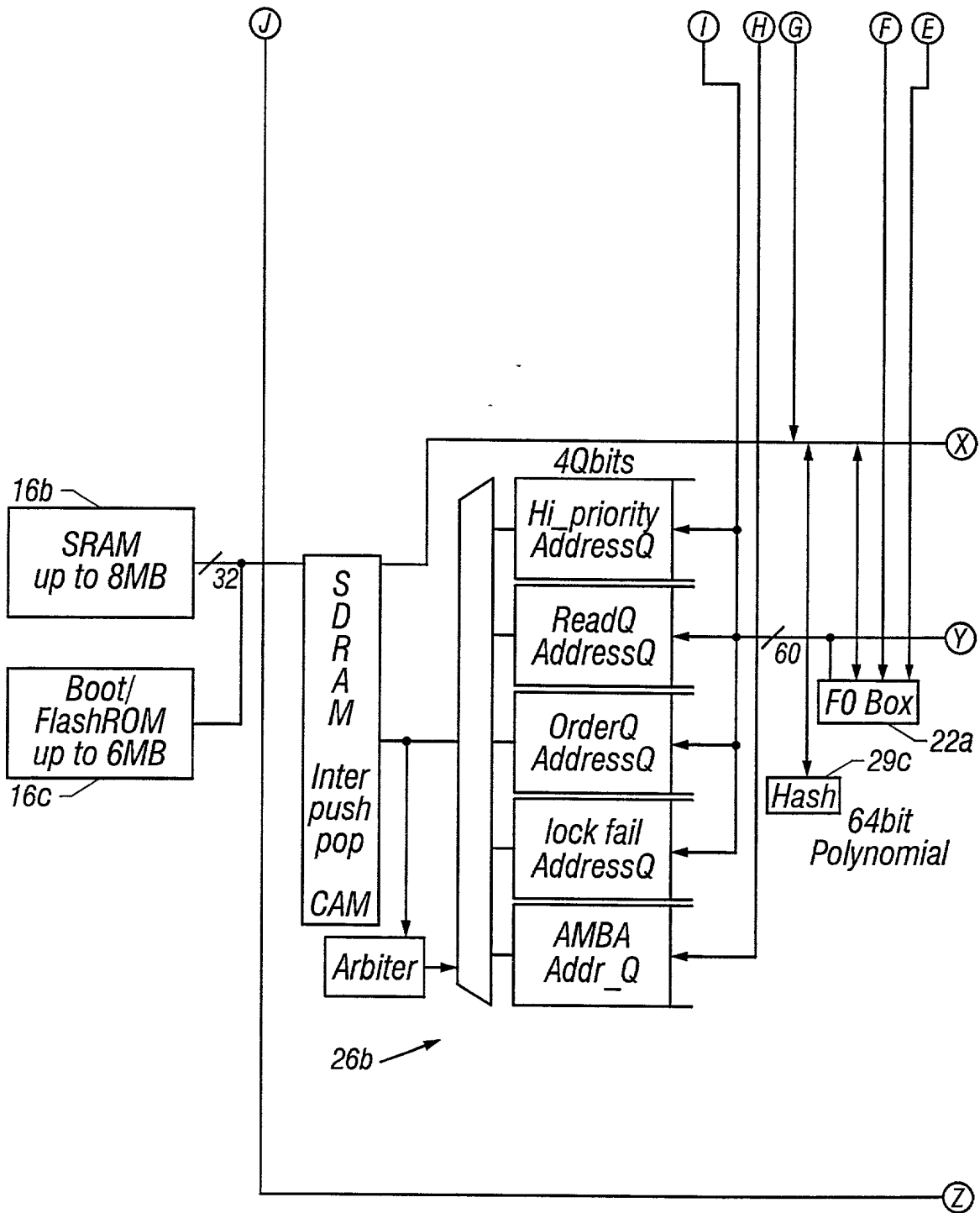


FIG. 2-3

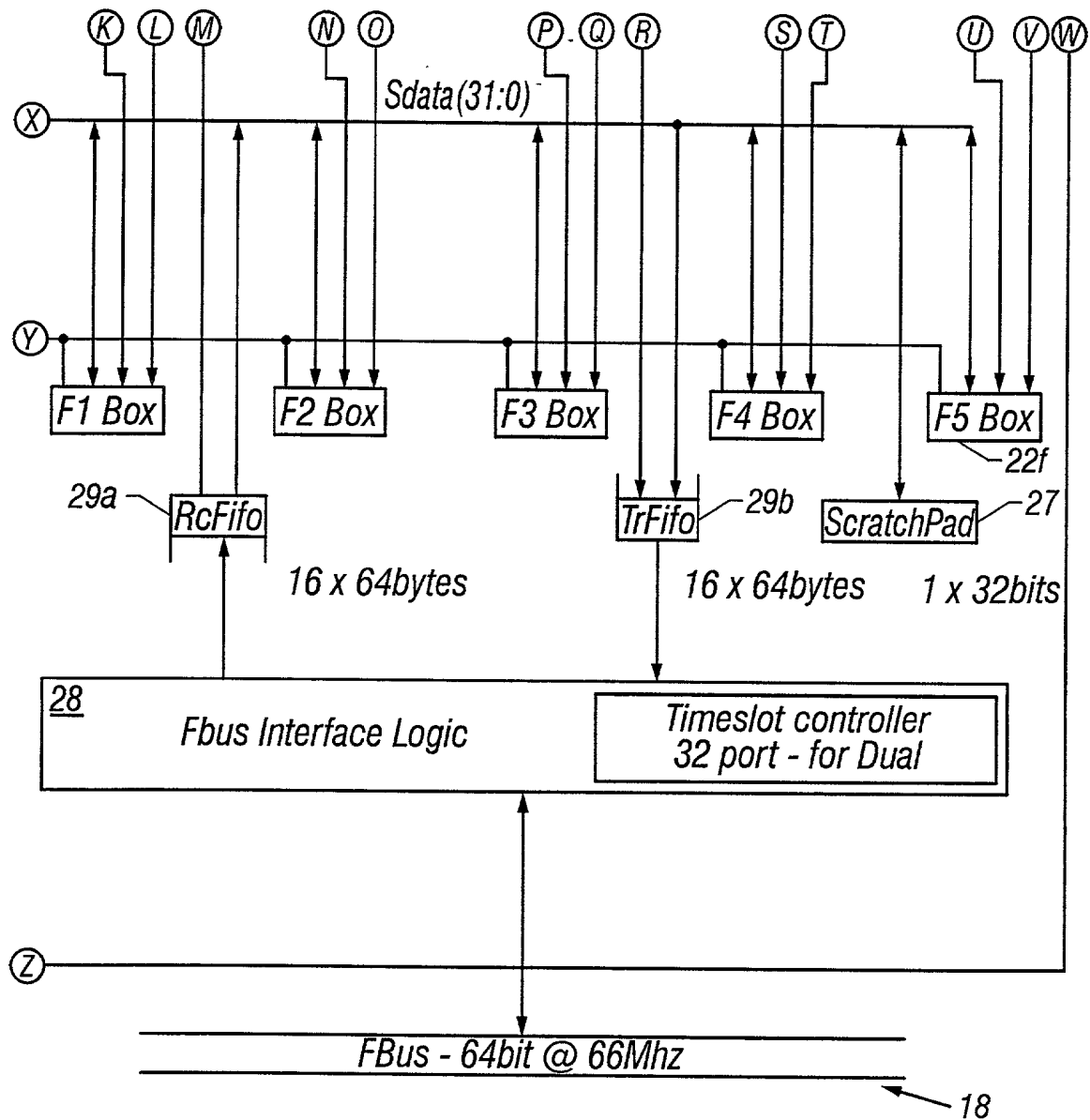


FIG. 2-4

6/20

AMBAI31:01

MBUSI31:01

SBUSI31:01

(A)

(B)

(C)

(D)

(E)

(F)

22

SEQ#\_event\_response  
FBI\_event\_response  
sram\_event\_response  
sdram\_event\_response  
amba\_event\_response

Context Event Arbiter

74

uengine controller

72

72c

uPC\_1

72b

uPC\_2

uPC\_3

uPC\_4

72a

73

decode

immediate\_data

address

control store  
1024 words

70

32bit words

FIG. 3-1

09867064-05001

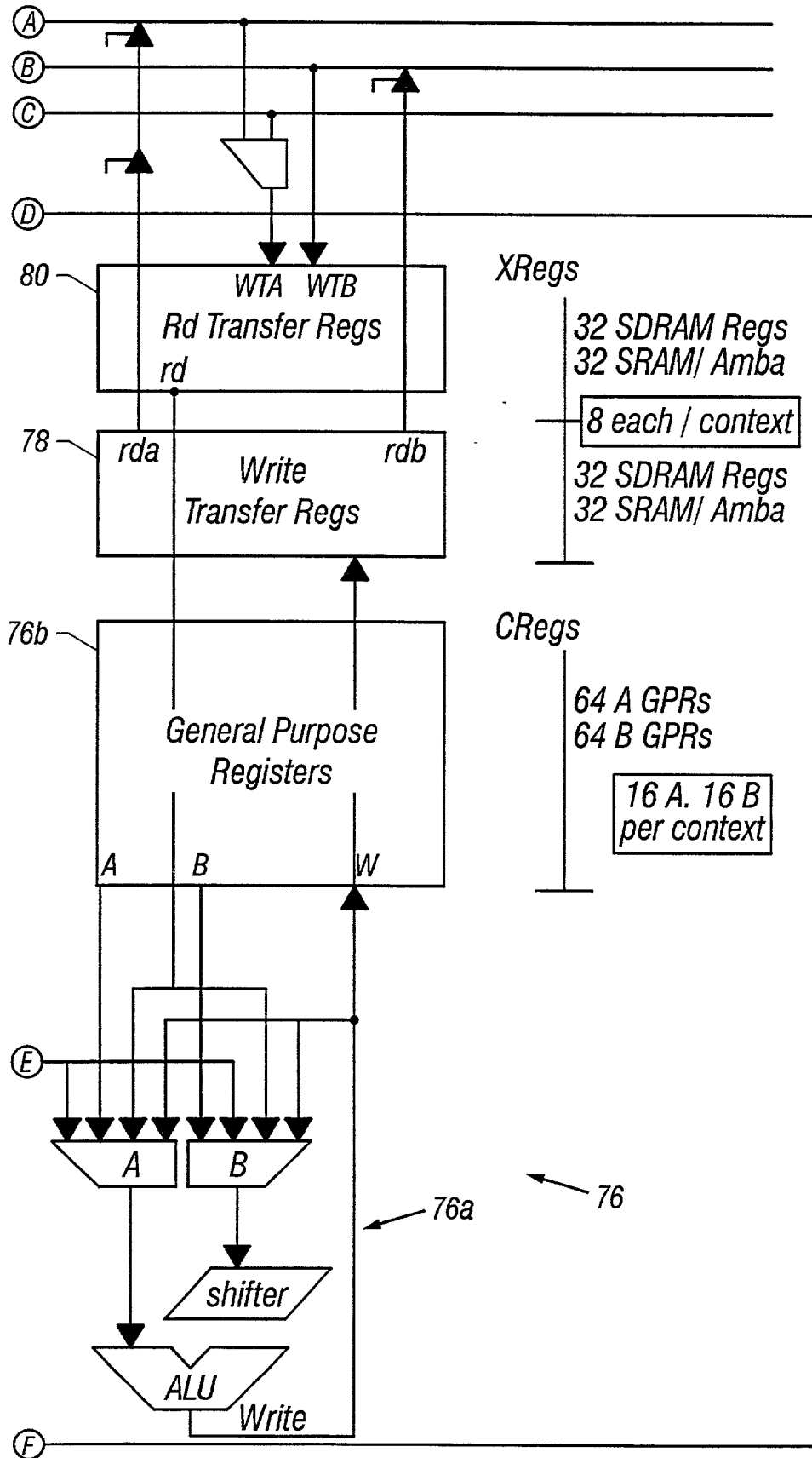
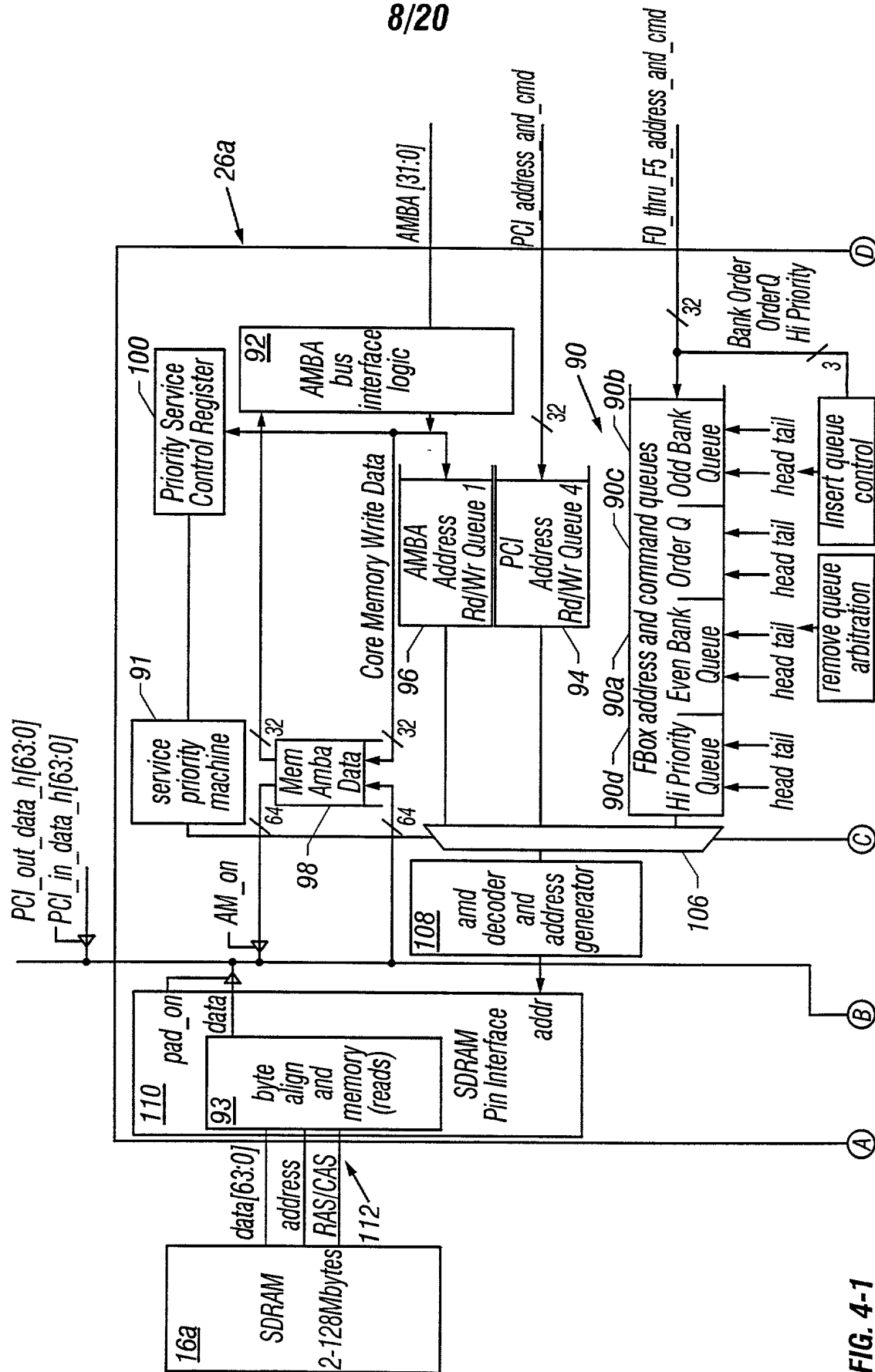


FIG. 3-2

09357064-052901





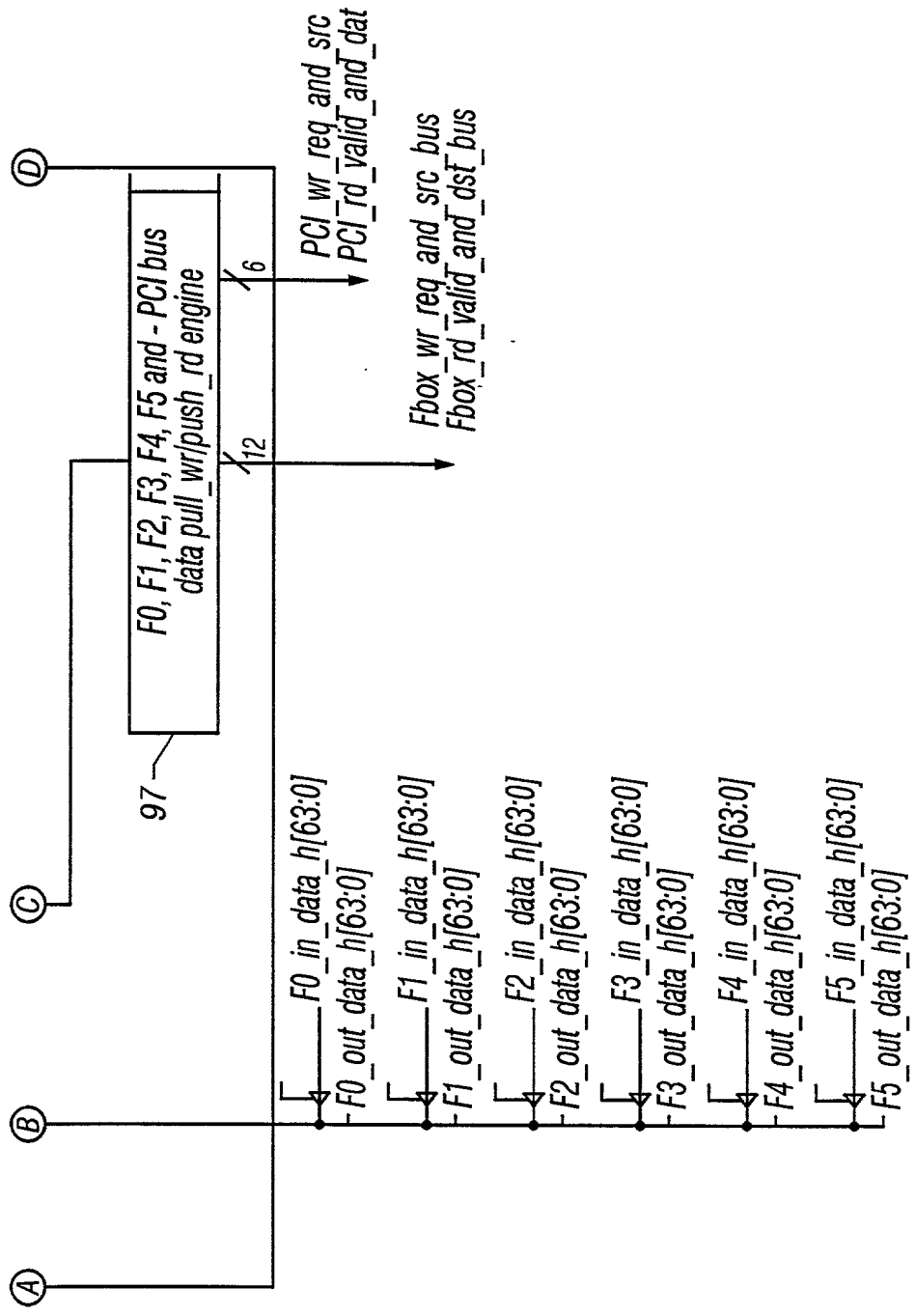


FIG. 4-2

10/20

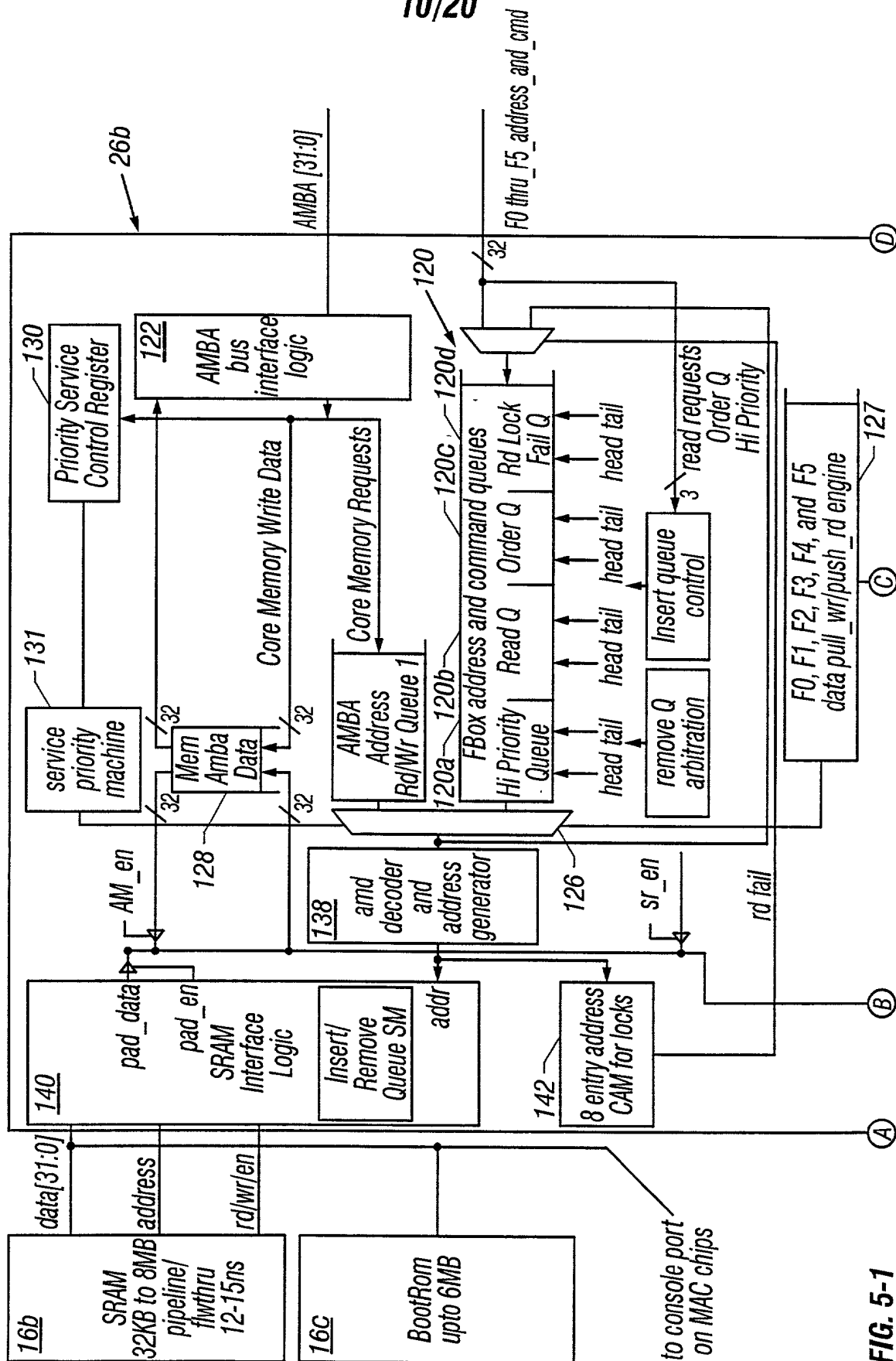
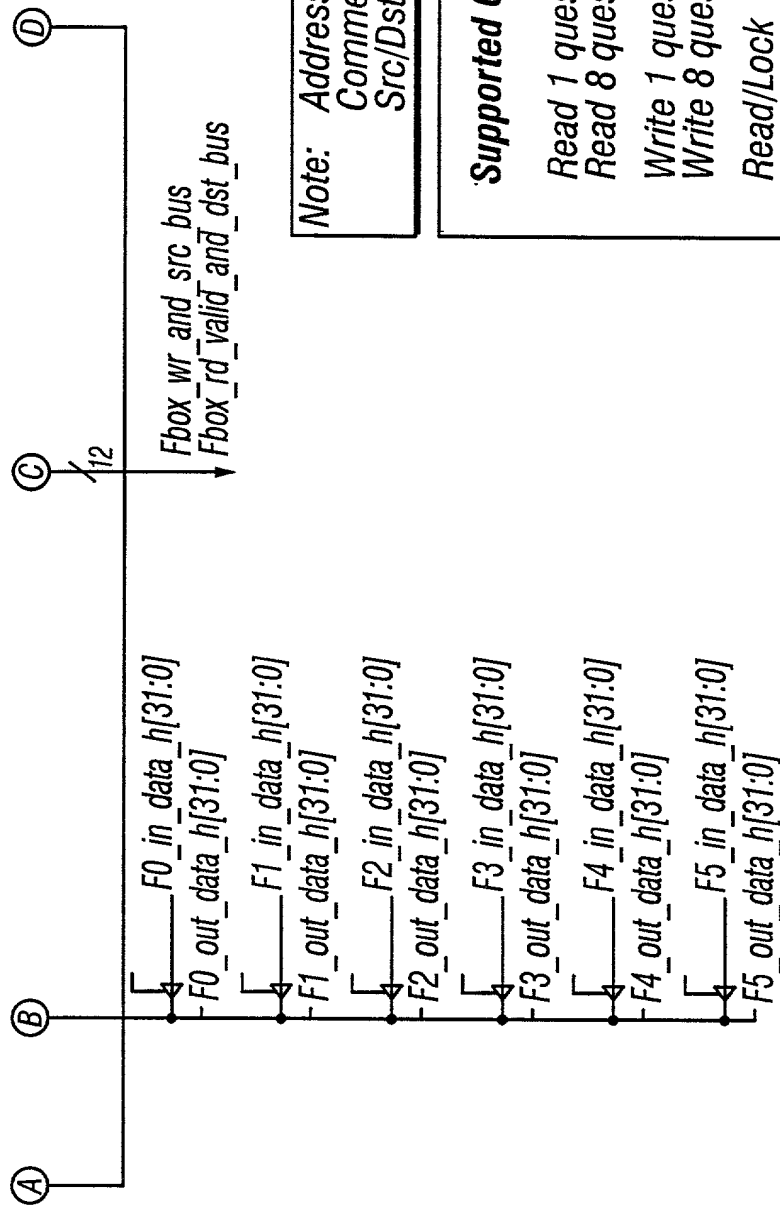


FIG. 5-1



Note: Address: 22 bits  
 Comment: 4 bits  
 Src/Dst: 6 bits

**Supported Comments:**

Read 1 questword  
 Read 8 questwords  
 Write 1 questword  
 Write 8 questwords  
 Read/Lock  
 Write/Unlock  
 Unlock  
 Reserve n locks  
 Insert Queue Element  
 Remove Queue Element

FIG. 5-2

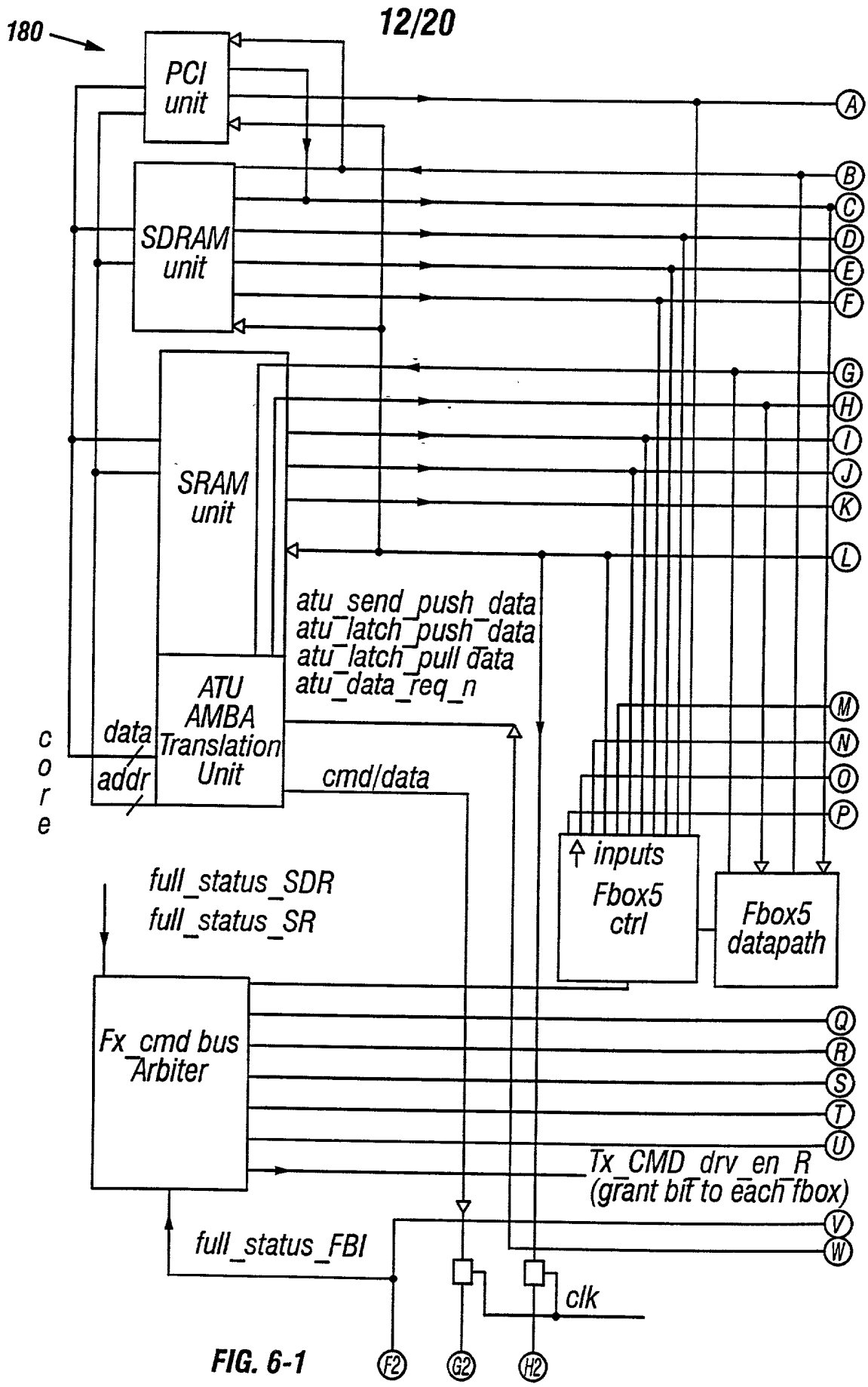


FIG. 6-1

F2 G2 H2

FIG. 6-1 12/20 180

13/20

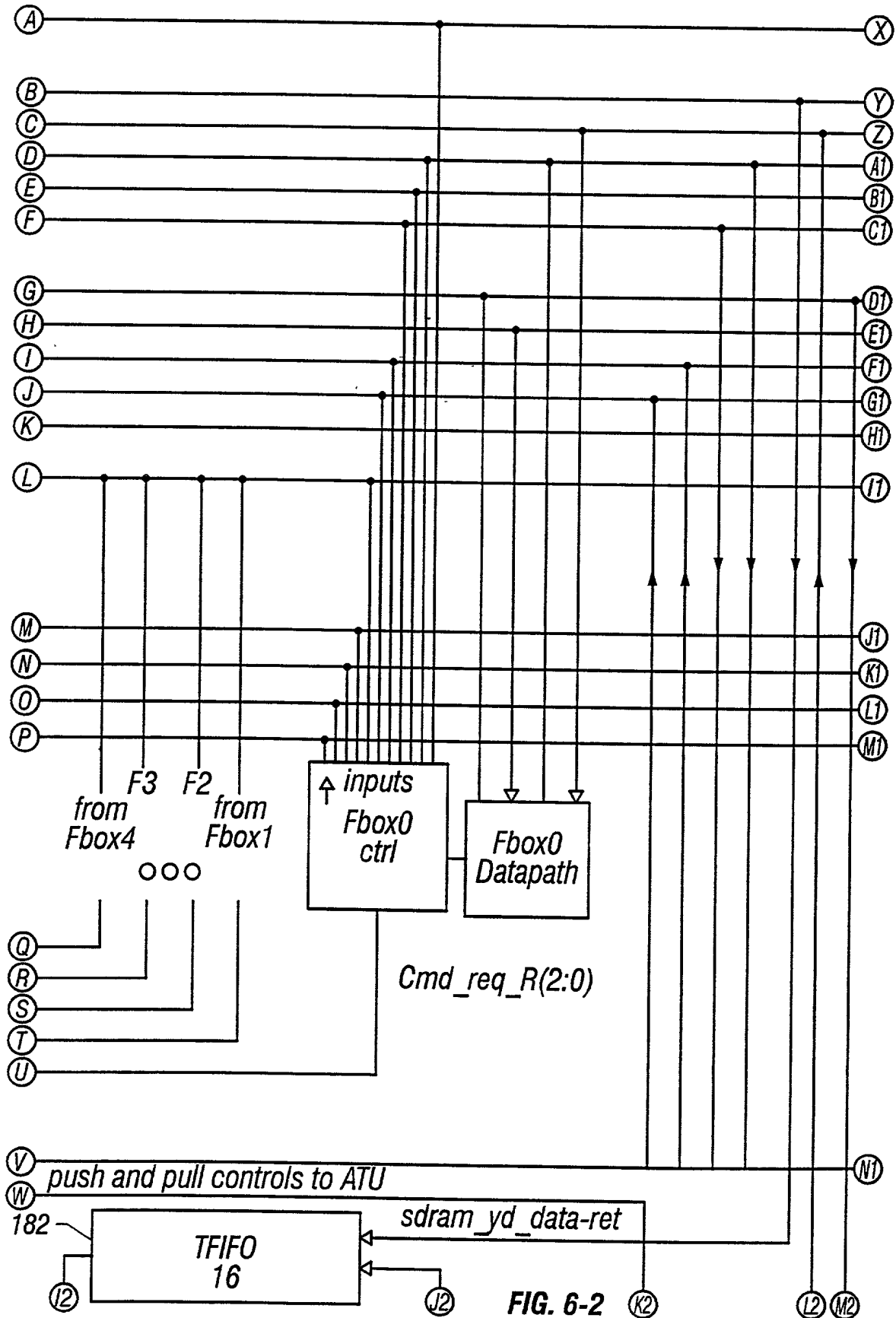


FIG. 6-2

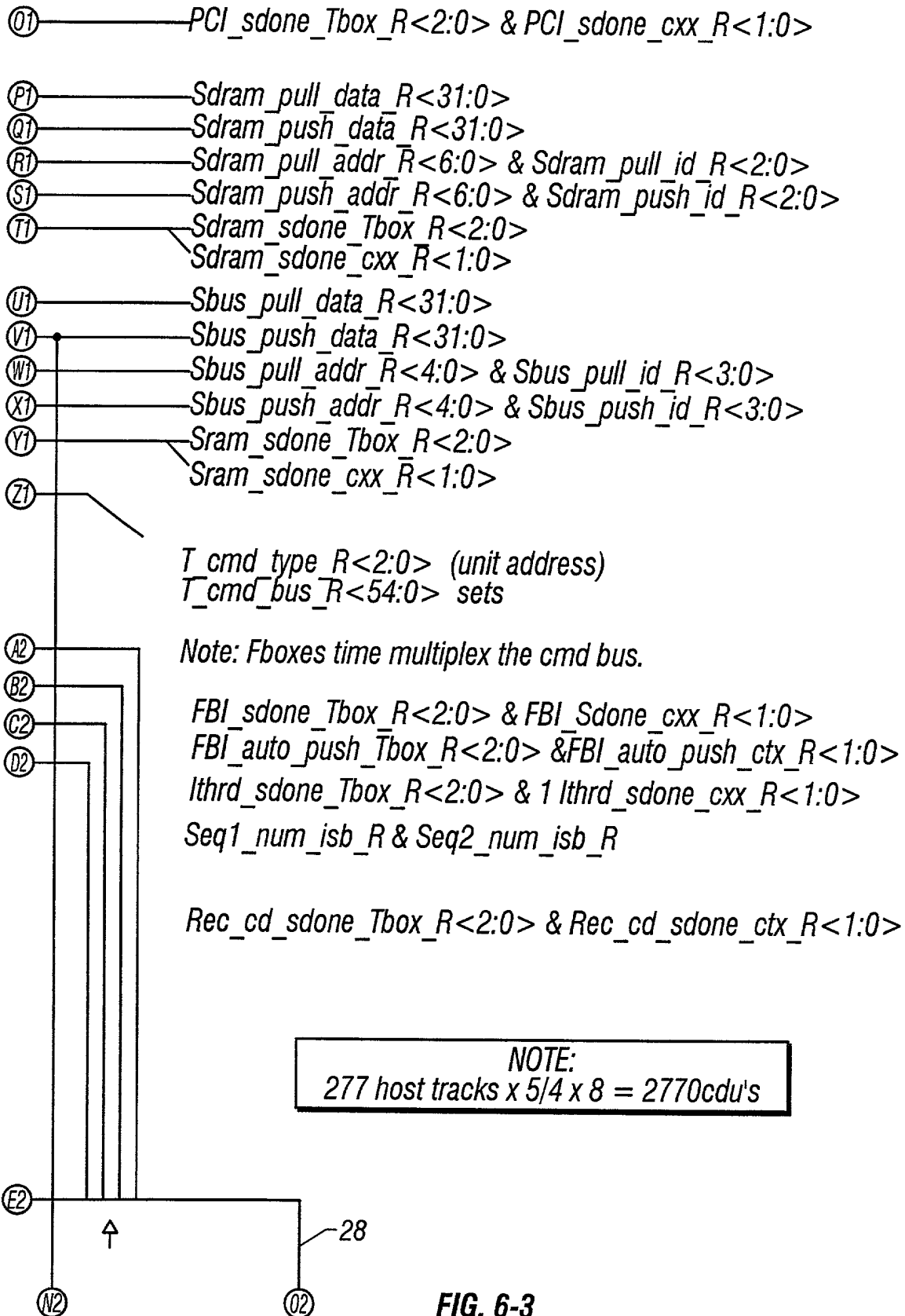
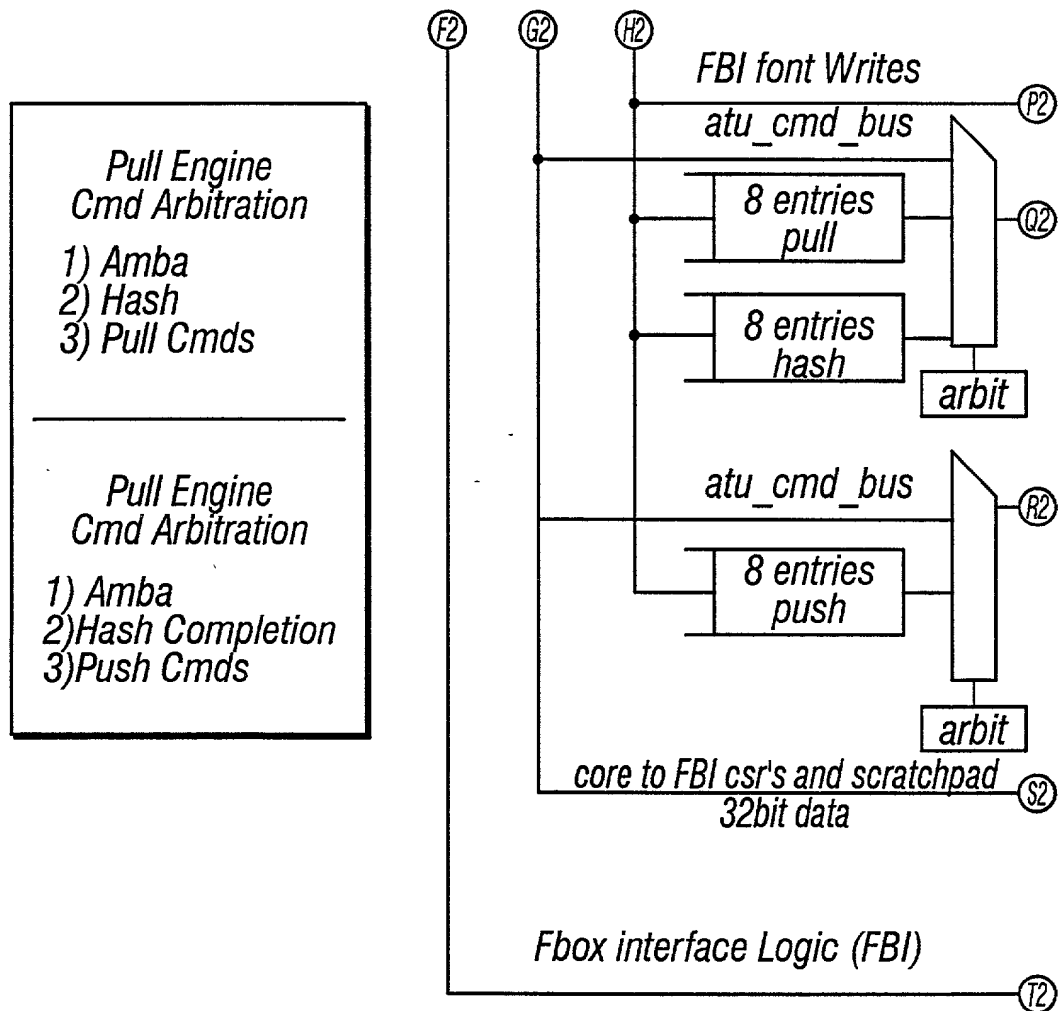


FIG. 6-3



#### ATU Notes:

- a) Core to FboxRegs:  
use sram\_push\_data\_bus
- b) Core to FBI Regs:  
use private ATU/FBI  
cmd/data bus
- c) Core reads FboxRegs:  
use SRAM\_pull\_data\_bus
- d) Core reads FBIRegs:  
use sram\_push\_data\_bus  
(makes sram appear like  
another Fbox to FBI on  
sram\_push\_bus)

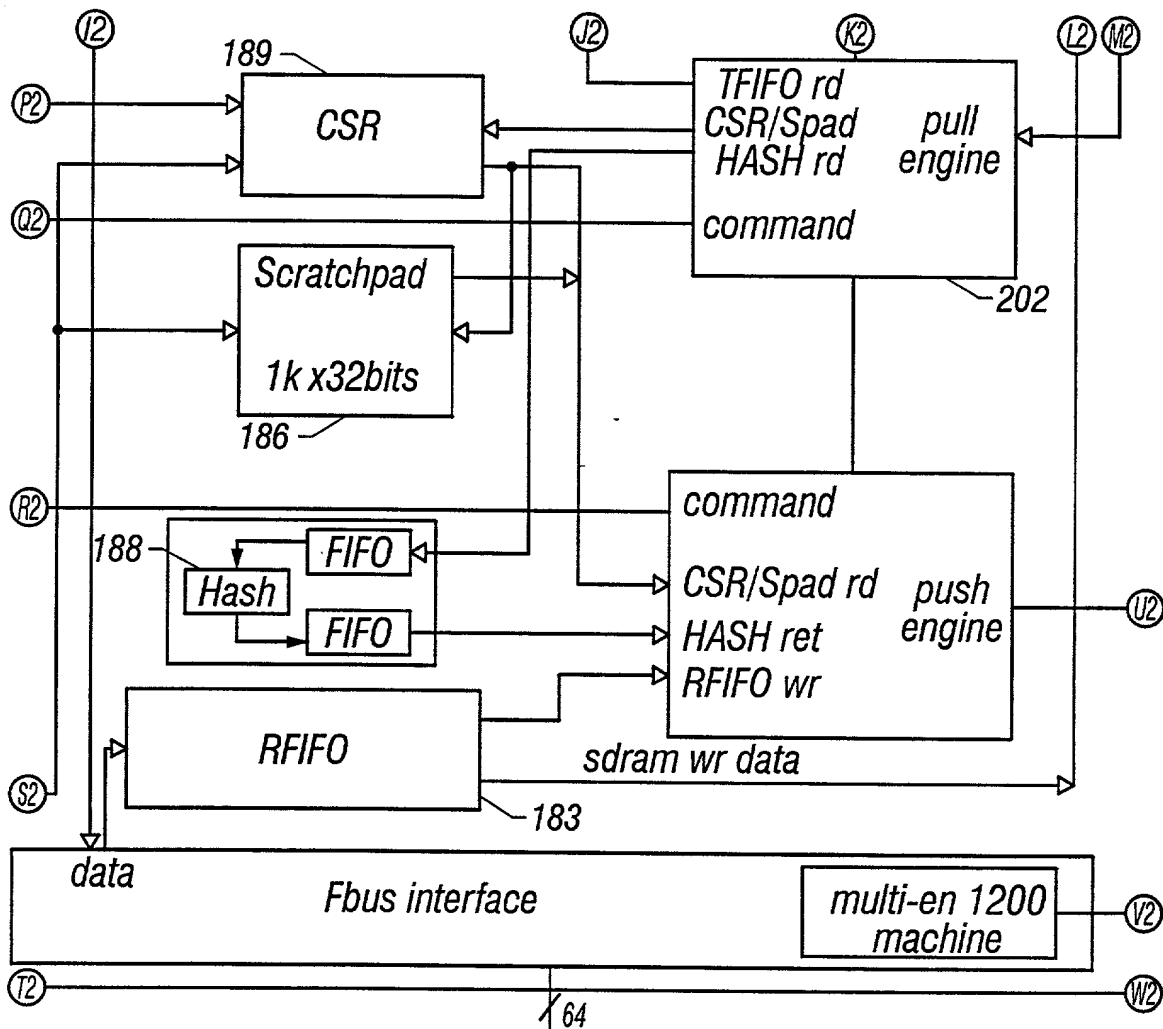
#### Cmd\_Req\_R<2:0>

000 none  
001 Sram Chain  
010 SDR chain  
011 Sram  
100 SDR  
101 FBI  
110 PCI  
111

#### Tx\_CMD\_drv\_en\_R<1:0>

0 none  
1 grant

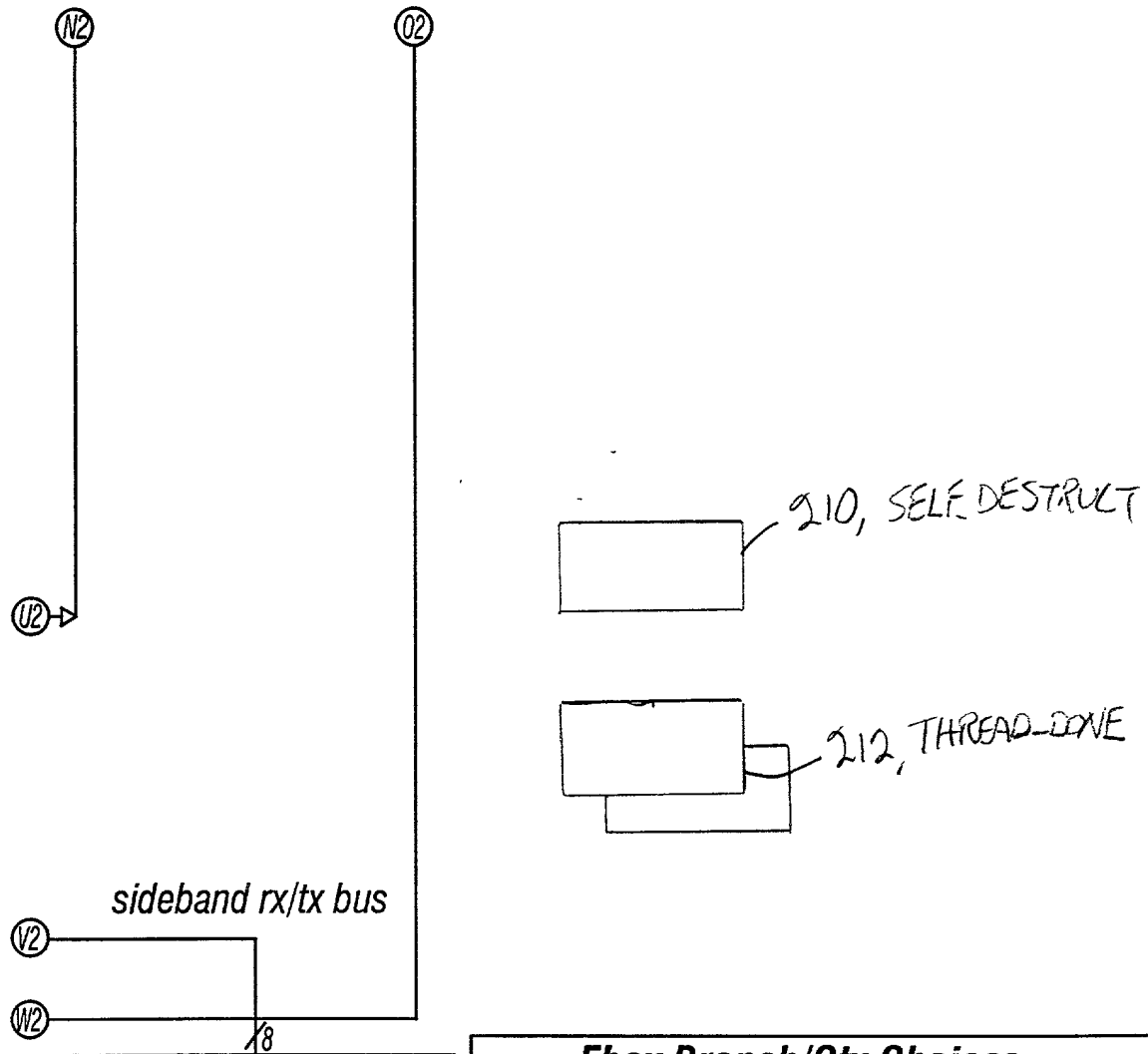
FIG. 6-4



$Sdram\_puXX\_addr\_R<6:0>$   $Sram\_puXX\_addr\_R<4:0>$   
 $[4:0]$  xfer\_reg\_addr  $[4:0]$  xfer\_reg\_addr  
 if not TFIFO  
 $[6:0]$  TFIFO\_addr  
 $Sdram\_puXX\_ID\_R<3:0>$   $Sram\_puXX\_ID\_R<3:0>$   
 0-5 Fboxes 0-5 Fboxes  
 8-13 Fboxes-csr 8-13 Fboxes-csr  
 6 fbi 6 fbi  
 15 nop 15 nop

FIG. 6-5





<i>Fbox Branch/Ctx Choices</i>	
1) FBI_sdone	br / ctx
2) FBI_auto_push	br / ctx
3) lthread_sdone	br / ctx
4) signal_rec_cxt	br / ctx
5) Seq#1_change (flag)	br / ctx
6) Seq#2_change (flag)	br / ctx
7)SRAM_sdone	br / ctx
8)SDRAM_sdone	br / ctx
9) volunteer_cxx_swap	ctx
10) Rec_req_available (flag)	br
11)SDRAM rd parity en (flag)	br
12) Fbox_push_protect	br
13) ccodes, contexts and kill	

FIG. 6-6

18/20

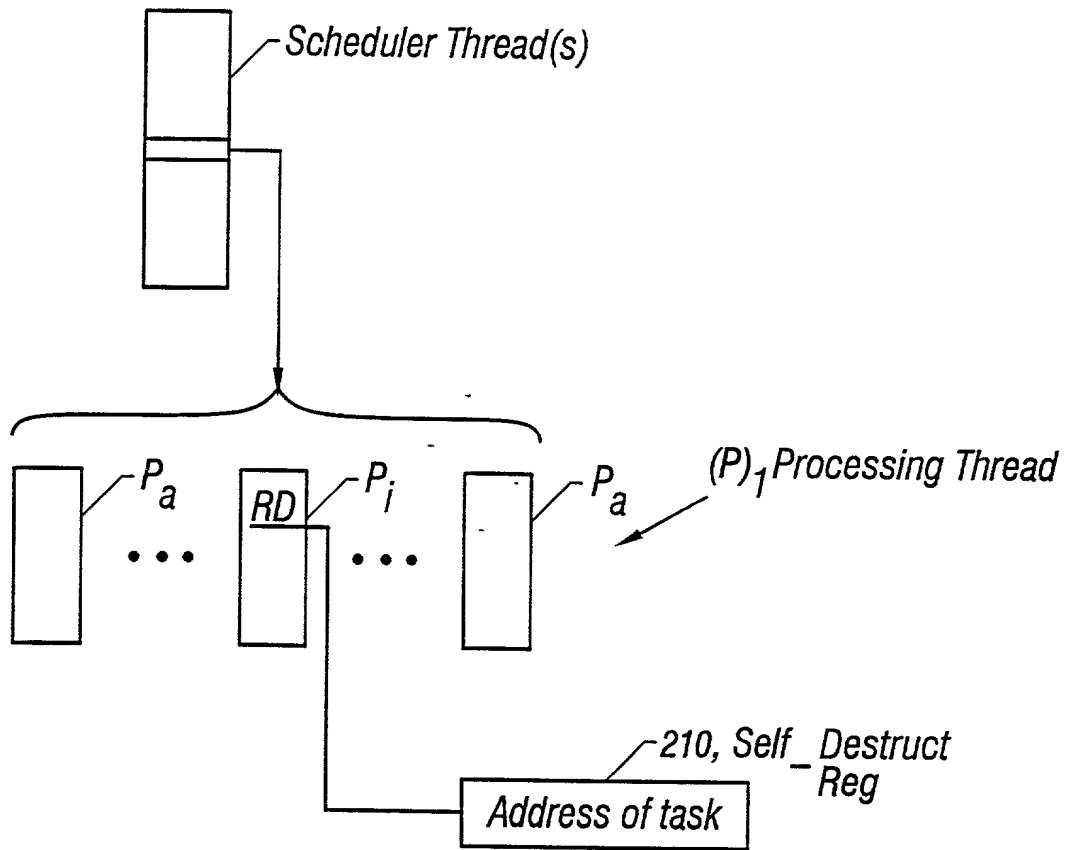


FIG. 7A

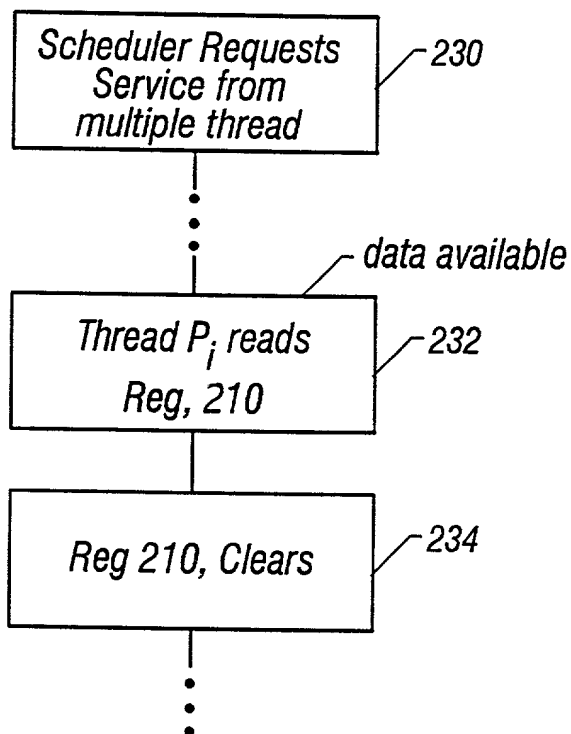


FIG. 7B

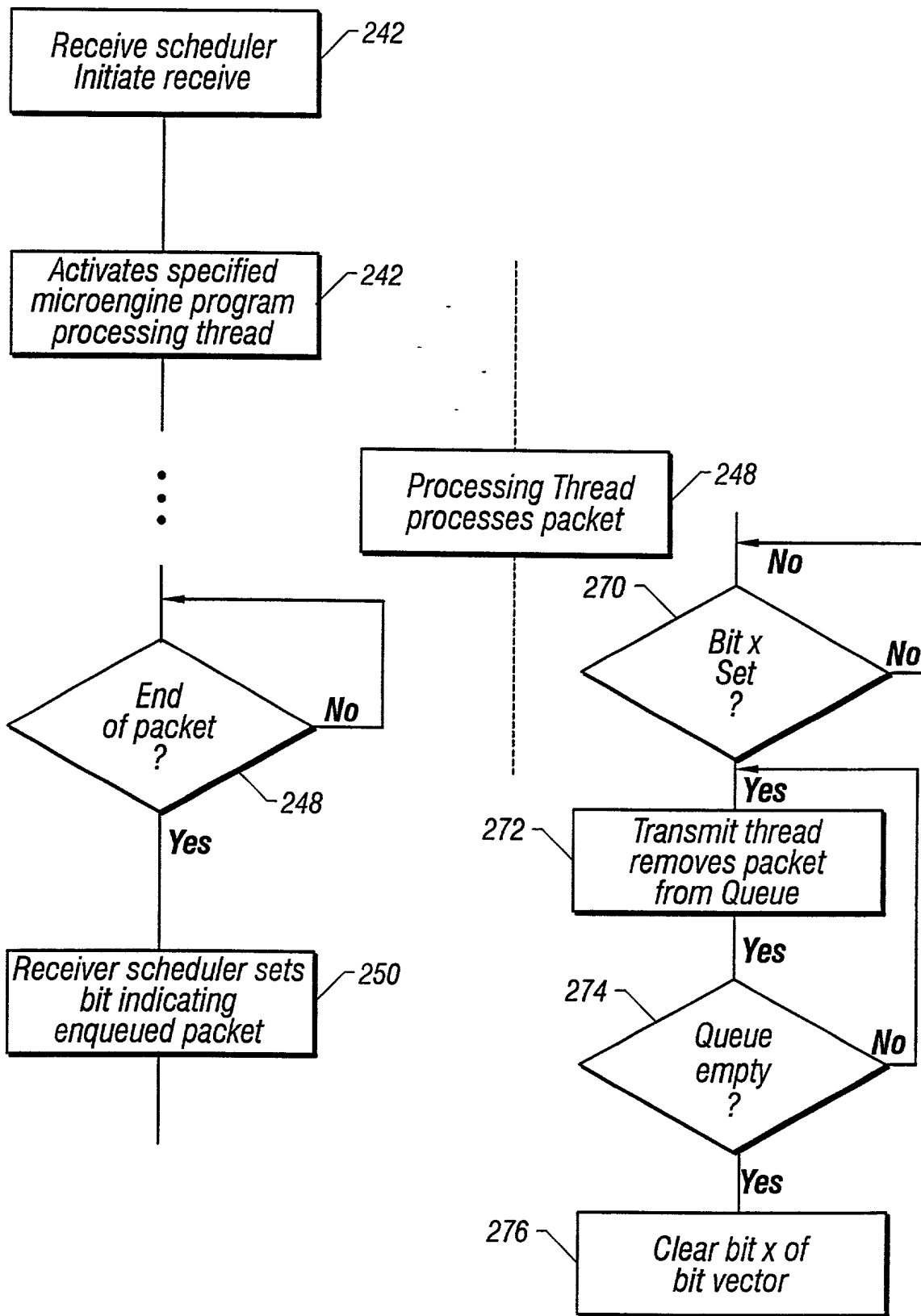


FIG. 8

20/20

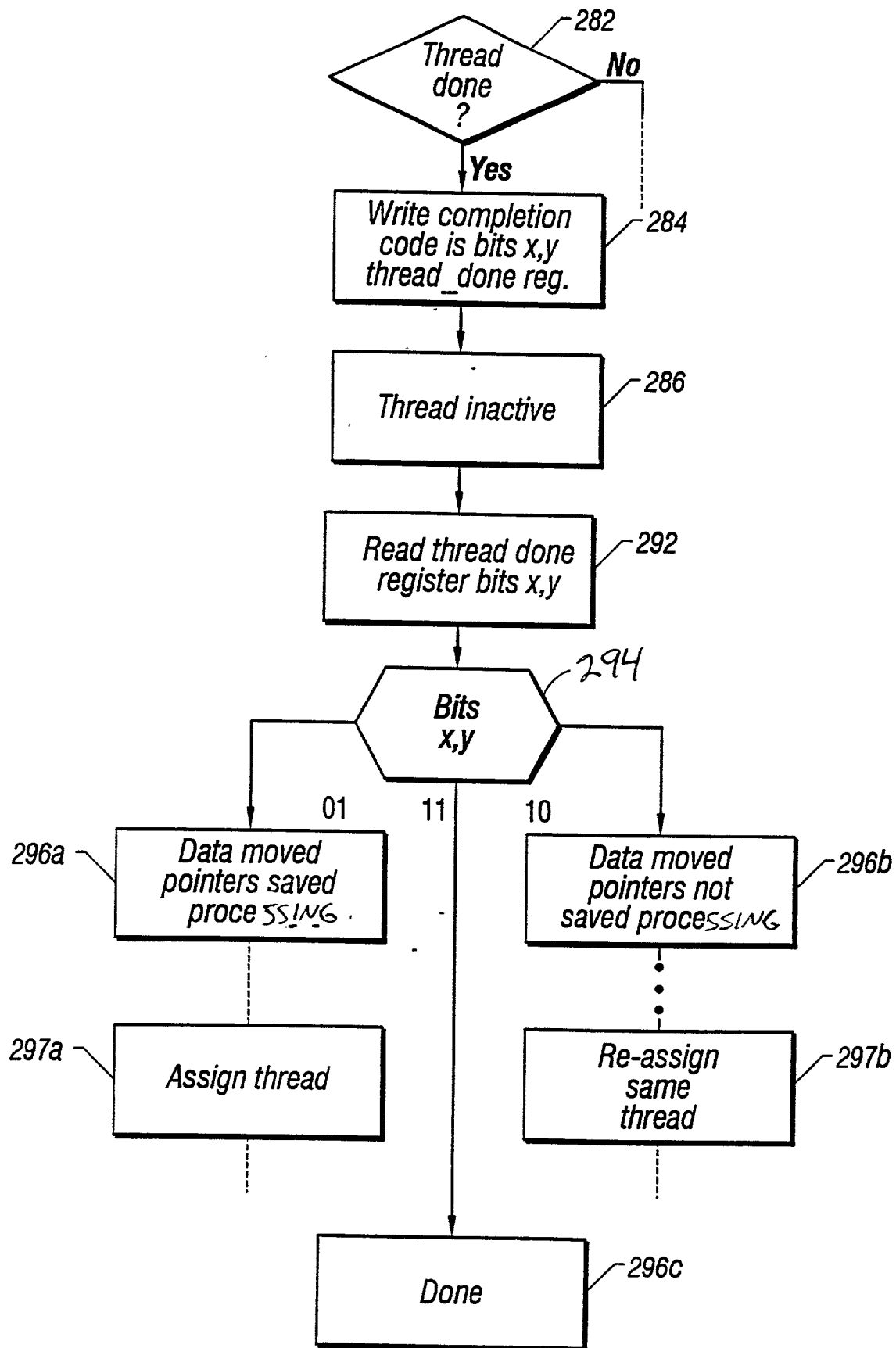


FIG 9

09867064-05901  
T05250-49079860